LGNet: Local-and-Global Feature Adaptive Network for 3D Interacting Hand Mesh Reconstruction

Haowei Xue[1,2,3] Meili Wang[1,2,3,*]

***Abstract*— Accurate 3D interacting hand mesh reconstruction from RGB images is crucial for applications such as robotics, augmented reality (AR), and virtual reality (VR). Especially in the field of robotics, accurate interacting hand mesh reconstruction can significantly improve the accuracy and naturalness of human-robot interaction. This task requires accurate understanding of complex interactions between two hands and ensuring reasonable alignment of the hand mesh with the image. Recent Transformer-based methods directly utilise the features of the two hands as input tokens, ignoring the correlation between local and global features of the interacting hands, leading to hand ambiguity, self-obscuration and self-similarity problems. We propose LGNet, Local and Global Feature Adaptive Network, by decoupling the hand mesh reconstruction task into three stages: a joint stage for predicting hand joints; a mesh stage for predicting a rough hand mesh; and a refine stage for fine-tuning the mesh image alignment using an offset mesh. LGNet enables high-quality fingertip-level mesh image alignment, effectively models the spatial relationship between two hands, and supports real-time prediction. Extensive quantitative and qualitative results on benchmark datasets show that LGNet outperforms state-of-the-art methods in terms of mesh accuracy and image alignment, and demonstrates strong generalisation capabilities in experiments on in-the-wild images.The code and dataset will be publicly available at https://github.com/HaoWeiHsueh/LGNet**

I. INTRODUCTION

Recovering the 3D mesh of two interacting hand from a monocular view is one of the fundamental tasks towards various robotic applications, including augmented reality (AR), virtual reality (VR), human-computer interaction (HCI), holographic transmission, digital medicine, etc. The main challenge in recovering interacting hand mesh is to capture the context of their interactions, which is different from single hand mesh reconstruction. Existing methods [1], [2], [3], [4] use 2.5D heatmaps to estimate hand joint positions [1], [2], [3] or use them as attention maps to extract sparse image features [4], which make it difficult to simulate occlusion of the hand surfaces and to extract the dense interaction context information. Inspired by attention mechanisms, [5], [6], [7] employ Transformer [8] to capture the correlation between two hands. Hampali et al. [5] (Figure 1 (a)) directly use the features of the backbone as the input tokens for Transformer. Li et al. [6] and Di et al. [7] (Figure 1 (b)) separate the features from the backbone to left- and right-hand-specific features, and then use the separated features as input tokens for the Transformer, which ignores the correlation between local and global features of the interacting hands, leading to hand feature ambiguity problem. To address the hand feature ambiguity problem, we propose LGNet, a Local and Global feature adaptive network, which crucially decouples the 3D interacting hand mesh reconstruction process into three stages: 1) The joint stage encodes the input image and predicts the hand joints. 2) The mesh stage quickly predicts the rough 3D hand mesh. 3) The refine stage quickly aligns the rough mesh with the user's hand image. Figure 1 (c) shows the LGBlock driven by our Transformer-based module. Quantitative and qualitative experiments demonstrate the effectiveness of LGNet. On the challenging InterHand 2.6M dataset, LGNet significantly outperforms existing methods and is suitable for real-time applications, generating well-aligned two-handed results on in-the-wild images and live video streams, as shown in Figure 1. Our contributions are summarized as follows:

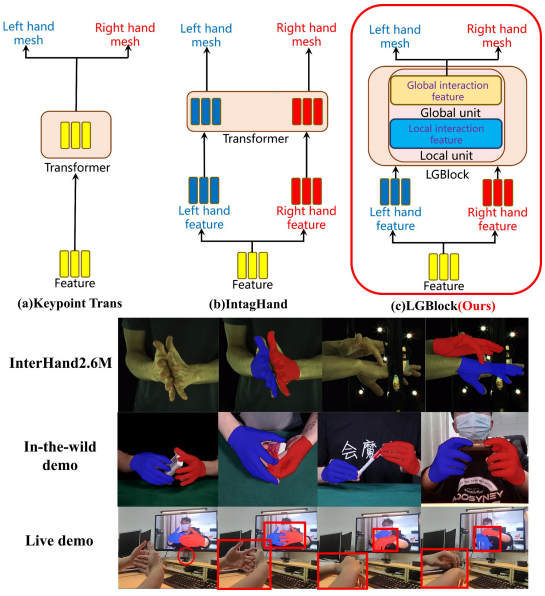

Fig. 1. The comparison between the previous Transformer blocks and LGNet, and hand reconstruction results are shown.

- We propose the LGNet, a local-and-global feature adaptive network for 3D interactive hand mesh reconstruction. Decoupling the hand mesh reconstruction process into joint, mesh, and refine stage, which can efficiently model the context of two-hand interaction, and can achieve fingertip-level mesh image alignment to support

*Corresponding author: wml@nwsuaf.edu.cn
[1]College of Information Engineering, Northwest A&F University, Yangling 712100, China
[2]Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture, Yangling 712100, China
[3]Shaanxi Key Laboratory of Agricultural Information Perception and Intelligent Service,Yangling 712100, China

robotics, VR/AR applications.
- We design the LGFA. LGFA-driven Local unit and Global unit to extract global interaction features and adapt them to each hand. The joint features fused with global information provide disambiguation cues for local visual features, which solves the hand feature ambiguity problem.
- Quantitative and qualitative experiments show that LGNet outperforms existing state-of-the-art methods in 3D hand mesh benchmarking.

II. Related Work

3D Interacting Hand Mesh Reconstruction. 3D interacting hand mesh reconstruction is one of the key challenges in robotics applications. [9], [10], [11] fit parametric hand models by optimising the energy function, but tend to fall into local optimal solutions and are computationally expensive. [12], [13], [14], [15], [16], [17] can handle each hand individually, but is unable to handle close hand interactions. It is currently common to train deep neural networks such as segmentation [18], [19], pixel-mesh correspondence maps [19], [20], or dense relative depth maps of interacting hand [20] to reduce the search space for hand poses and the difficulty of optimising the energy function. However, this hybrid method cannot be trained end-to-end and the optimisation process tends to fall into local minima. As a result, it is difficult for existing methods to achieve accurate alignment with images, which is crucial for robotics and VR/AR applications. To this end, we decouple the hand mesh reconstruction process so that each stage focuses on a specific task, improves network learning efficiency, and achieves a lightweight architecture. Our method generates 3D hand meshes that are accurately aligned with the user's hand in VR/AR views and achieves real-time performance.

3D Interacting Hand Pose Estimation on InterHand2.6M dataset. InterHand2.6M dataset [3], [21] is the first interactive hand mesh dataset. Following the release of the dataset, Moon et al. [3] extended the one-handed pose estimation method to two-handed interaction scenarios while predicting 2.5D heatmaps for both hands. Zhang et al. [4] propose to estimate a 3D hand mesh based on the 2.5D heatmaps. Rong et al. [22] propose a two-stage framework to alleviate the hand collision problem between hands. Meng et al. [23] employ an erasure mechanism to convert a two-handed image into two single-handed images. However, these methods fail to adequately model the dependency between the two hands. Transformer-based methods [6], [5] hdirectly use the features of two hands as input tokens, ignoring the correlation between the local and global features of the interacting hands, leading to the hand feature ambiguity problem, and the model is difficult to model complex hand spatial relationships. In contrast, we propose LGNet to solve the hand feature ambiguity problem.

III. Method

Figure 2 shows the overall structure of our LGNet and the three-stage framework for recovering an interacting 3D hand mesh from a single RGB image:

- **The joint stage** encodes the input image and predicts the 2.5D joint coordinates ($\mathbf{J_R}$ or $\mathbf{J_L}$) and joint features ($\mathbf{F_{JR}}$ or $\mathbf{F_{JL}}$) of each hand.
- **The mesh stage** enhances the features ($\mathbf{F_{JR}}$ or $\mathbf{F_{JL}}$) of the joint stage and predicts a rough hand mesh $\mathbf{M_r}$.
- **The refine stage** generates the final hand mesh ($\mathbf{M_f} = \mathbf{M_r} + \Delta\mathbf{M}$) by regressing the offset mesh ($\Delta\mathbf{M}$) through a graph convolutional network (GCN).

A. Joint Stage

1) Feature extractor: Given an input hand image $\mathrm{I} \in \mathbb{R}^{H\times W\times 3}$ ($H = 256$,$W = 256$), the joint stage begins with encoding it using a feature extractor. We use ResNet-50 [24], which has been pre-trained on ImageNet [25], as the backbone to extract the overall features of the image $\mathrm{F} \in \mathbb{R}^{h\times w\times C}$($h = H/32$, $w = W/32$, $C = 2048$). Then, we convert the image features $\mathbf{F}$ into right hand features $\mathbf{F_R}$ and left hand features $\mathbf{F_L}$ through two independent 1×1 convolutional layers. These two features have the same dimension $\mathbb{R}^{h\times w\times c}(c = C/4$).

2) Local-and-Global feature adaptive block (LGBlock) : Figure 3 shows the detailed flow of LGBlock. LGBlock consists of two steps, Step 1 local unit and Step 2 global unit, each using the LGFA module.

- Step1. The local unit for hand local interaction feature extraction. The blue part of the Figure 3 shows the overall flow of the local unit of LGBlock.
- Step2. The global unit for hand global interaction feature adaptation. The yellow part of the Figure 3 shows the overall flow of the global unit of LGBlock.

Local unit: Graph Convolution for Two-Hand Modeling. We design and implement a novel GCN module that extends single-hand processes into two-hand processes. We introduce residual connections after every two GINConvBlock operations to facilitate gradient propagation and enhance learning. See Figure 3, which describes the framework of Graph-CNN, where the network starts with a GINConvBlock containing a GIN convolutional layer with ReLU activation and one-dimensional batch normalization. Subsequently, the network employs three GINResBlocks, each consisting of two GINConvBlocks and an identity connection, and finally a GINConvBlock. We use the fully-connected (FC) layer FC($\cdot$) to convert $\mathbf{F}$ into a feature vector FC($\mathbf{F}$), which is shared among all vertices. We then combine the dense matching encoding $\mathrm{d_i}$ of the $\mathrm{i^{th}}$ vertex with the shared vectors to generate the feature $\mathbf{F_V^i}$ for each vertex. The process can be represented as:

$$\mathrm{F_V^i} = \mathrm{Concat}(\mathrm{FC}(\mathrm{F}), \mathrm{d_i}), \mathrm{i} = 0, 1 \ldots, \mathrm{N_0}, \tag{1}$$

In this process, each vertex $\mathbf{F_V^i} \in \mathbb{R}^f$ represents the initial feature vector, where the feature length is $f = 512$. For the rough sub-mesh, it contains $\mathrm{N_0} = 63$ vertices. By stacking $\mathbf{F_V^i}$, we obtain $\mathbf{F_V^t} \in \mathbb{R}^{N\times f}$, where $\mathrm{t} = 0$. Subsequently, we perform Chebyshev spectral graph CNN operations at each

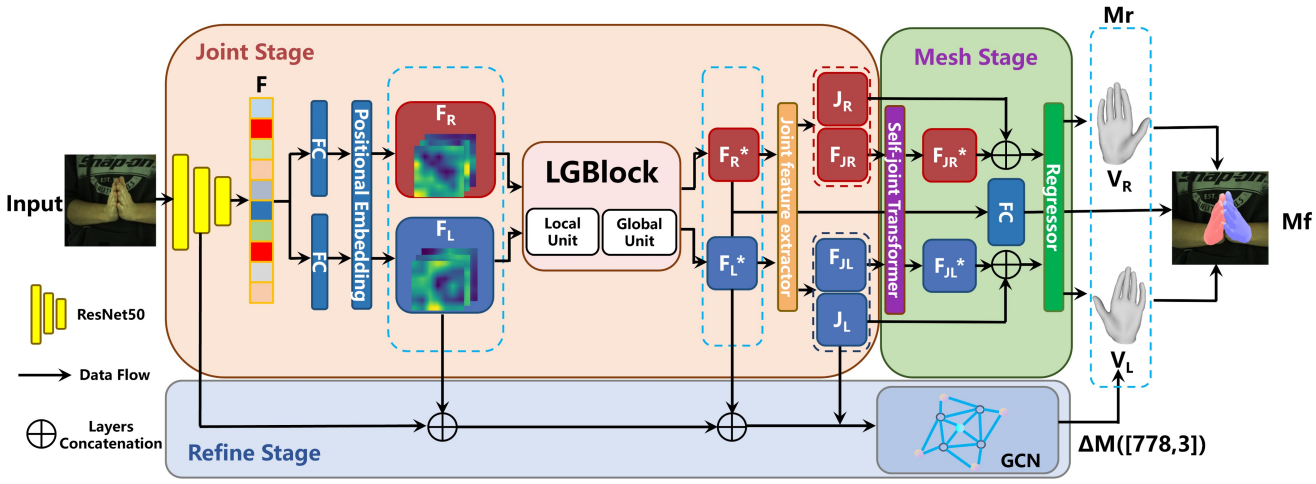

Fig. 2. The overall architecture of Local-and-Global Feature Adaptive Network (LGNet).

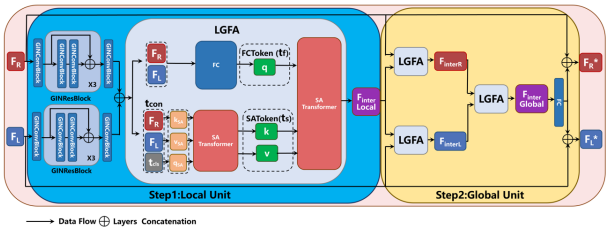

Fig. 3. The overall architecture of Local-and-Global Feature Adaptive Block (LGBlock).

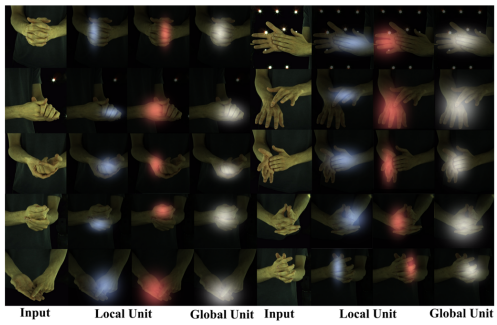

Fig. 4. Visualization of attention maps in pseudo color.

t-th ($t = 0, 1, 2$) block to convert the input vertex features $\mathbf{F_V^t}$ into $\mathbf{F_{GCN}^t}$.

$$F_{GCN}^t = NAF(F_V^t(\widehat{L}^t)\sum_{k=0}^{K-1} T_k^t W_k^t), \tag{2}$$

$\widehat{L}^t$ denotes the scaled Laplacian matrix, T_k^t is the k-th term of a Chebyshev polynomial of order K, W_k^t stands for the learnable parameters, and NAF$(\cdot)$ denotes the nonlinear activation function. $\mathbf{F_{GCN}^t}$ represents the intermediate features transmitted to the LGBlock module.

Local unit: Local-and-Global Feature Adaptive Module(LGFA). LGFA innovatively converts the left-hand and right-hand features into two new tokens FCToken and SAToken as input tokens. We extract $\mathbf{q}$ from the FCToken, the FCToken $\mathbf{t_f} \in \mathbb{R}^{hw \times c}$ is generated as follows:

$$t_f = FC(\xi(\mathbf{F_R}, \mathbf{F_L})) \tag{3}$$

where $\xi(\cdot)$ denotes the combination of the connectivity function and the remodeling function, and FC$(\cdot)$ denotes the fully connected layer. We extract $\mathbf{k}$ and $\mathbf{v}$ from the SAToken, the SAToken $\mathbf{t_s} \in \mathbb{R}^{l \times c}$ is generated as follows:

$$\text{Attn}(\mathbf{q_{SA}}, \mathbf{k_{SA}}, \mathbf{v_{SA}}) = \text{Softmax}(\frac{\mathbf{q_{SA}}\mathbf{k_{SA}}^T}{\sqrt{d_{k_{SA}}}})\mathbf{v_{SA}}, \tag{4}$$

$$\mathbf{a} = \text{Attn}(\mathbf{q_{SA}}, \mathbf{k_{SA}}, \mathbf{v_{SA}}) + \mathbf{t_{con}}, \tag{5}$$

$$\mathbf{t_s} = \text{MLP}(\mathbf{a}) + \mathbf{a}, \tag{6}$$

where $d_{k_{SA}}$ denotes the channel dimension of $\mathbf{k_{SA}}$. Therefore, $\mathbf{t_s}$ is built from two hand local interaction features on the standard SA transformer. Finally, the SA transformer will output the local interaction features $\mathbf{F_{inter}^{local}}$ with $\mathbf{q}$, $\mathbf{k}$ and $\mathbf{v}$ according to Equation (4), Equation (5) and Equation (6). Therefore, in the local unit, LGFA will extract the local interaction feature $\mathbf{F_{inter}^{local}}$ by fusing two hand features ($\mathbf{F_R}$ and $\mathbf{F_L}$). Formally,

$$\mathbf{F_{inter}^{local}} = \text{LGFA}(\mathbf{F_R}, \mathbf{F_L}; \mathbf{W_{local}}) \tag{7}$$

where $\mathbf{W_{local}}$ denotes the learnable weights of the LGFA in the local unit.

Global unit: Hand Global Interaction Feature Adaptation. The global unit is formally represented as:

$$\mathbf{F_{interR}} = \text{LGFA}(\mathbf{F_R}, \mathbf{F_{inter}^{local}}; \mathbf{W_{globalR}}) \tag{8}$$

$$\mathbf{F_{interL}} = \text{LGFA}(\mathbf{F_L}, \mathbf{F_{inter}^{local}}; \mathbf{W_{globalL}}) \tag{9}$$

$$\mathbf{F_{inter}^{global}} = \text{LGFA}(\mathbf{F_{interR}}, \mathbf{F_{interL}}; \mathbf{W_{globalI}}) \tag{10}$$

where $\mathbf{W_{globalR}}$,$\mathbf{W_{globalL}}$,$\mathbf{W_{globalI}}$ are the learnable weights during the global feature adaptation process for right hand, left hand, and two hand interactions in LGFA. Eventually, we connect the adaptive global interaction features and their corresponding hand features ($\mathbf{F_R}$ or $\mathbf{F_L}$) along the channel

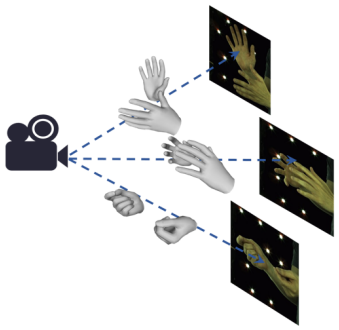

Fig. 5. High-resolution image features are extracted in the image feature space using bilinear interpolation by 3D-to-2D projection from the 3D hand mesh to the 2D image space, and image features are collected for each vertex in the rough hand mesh.

dimension, which become the final features for each hand, denoted by $\mathbf{F}_{\mathbf{R}}^{*}$ and $\mathbf{F}_{\mathbf{L}}^{*}$. It is shown in Figure 4 that the global unit also pays more attention to the closely interacting area, especially the finger-tips. This indicates that the LGBlock helps to address mutual collision between hands implicitly.

B. Mesh Stage

The Regressor generates 48-dimensional pose parameters (θ_{R} or θ_{L}) and 10-dimensional shape parameters (β_{R} or β_{L}) [26] for MANO [26] based on the enhanced hand joint features ($\mathbf{F}_{\mathbf{JR}}^{*}$ or $\mathbf{F}_{\mathbf{JL}}^{*}$). Finally, the final 3D hand mesh $\mathbf{M_r}$ (denoted by $\mathbf{V_R}$ or $\mathbf{V_L}$) is obtained by forwarding the MANO parameters to the MANO layer. In addition, the adaptive hand features ($\mathbf{F}_{\mathbf{R}}^{*}$ or $\mathbf{F}_{\mathbf{L}}^{*}$) are passed into the fully connected layer to output the 3D relative translation between the two hands.

C. Refine Stage

The blue part at the bottom of the Figure 2 shows the refine stage. To generate a fine hand mesh that is consistent with the real hand in the input image, each mesh vertex is projected into the two-dimensional image space of the input image through a three-dimensional to two-dimensional projection, as shown in Figure 5. Then, bilinear interpolation is performed around each projected vertex on the feature map to extract the associated feature vectors. Local and global features are extracted from the input image and the joint stage respectively, and all features are concatenated and sent to Graph-CNN. The goal of Graph-CNN is to estimate a three-dimensional offset vector for each mesh vertex based on the input three-dimensional coordinates and collected local and global features, in order to align the rough mesh with the hands in the image. We use the Graph Isomorphism Network (GIN) convolution regress the offset mesh ($\Delta\mathbf{M}$) to generate the final hand mesh ($\mathbf{M_f} = \mathbf{M_r} + \Delta\mathbf{M}$), which is defined as:

$$x'_i = \mathrm{MLPS}(x_i + \sum_{j \in N(i)} x_j), \tag{11}$$

where x_i denotes the features of the i-th node in the graph (equivalent to the i-th vertex in the rough mesh); x'_i denotes the update features of the node; N(i) is the set of indexes of the neighboring nodes of the ith node; MLP denotes a series of multilayer perceptrons.

D. Training & Inference

We denote a dataset as $\{I^i, J^i_{GT}, M^i_{GT}\}_{i=1}^{N}$, where N is the total number of samples in the dataset; I^i is the i-th input image; J^i_{GT} and M^i_{GT} are the three-dimensional joint and mesh coordinates of the hand. The three-dimensional joint coordinates can be computed from the mesh using a predefined regression matrix G($J^i_{GT} = GM^i_{GT}$). We use L1 loss to compute the mesh loss L_{mesh} and joint loss L_{joint} to supervise the predictions of the mesh and joints in the three stages:

$$L_{mesh} = \left\| M^i_{GT} - M^i_f \right\| + \left\| M^i_{GT} - M^i_r \right\|, \tag{12}$$

$$L_{joint} = \left\| J^i_{GT} - GM^i_f \right\| + \left\| J^i_{GT} - GM^i_r \right\|, \tag{13}$$

Additionally, we incorporate a normal loss L_{norm} to preserve surface normals and edge length loss L_{edge} to penalize flying vertices:

$$L_{norm} = \sum_{f \in M^i_{GT}} \sum_{e \in f} \left\| \left\langle \vec{e}_f, n^f_{GT} \right\rangle \right\| + \sum_{f \in M^i_{GT}} \sum_{e \in f} \left\| \left\langle \vec{e}_r, n^f_{GT} \right\rangle \right\|, \tag{14}$$

$$L_{edge} = \sum_{f \in M^i_{GT}} \sum_{e \in f} \left\| \left| \vec{e}_f \right| - \left| \vec{e}_{GT} \right| \right\| + \sum_{f \in M^i_{GT}} \sum_{e \in f} \left\| \left| \vec{e}_r \right| - \left| \vec{e}_{GT} \right| \right\|, \tag{15}$$

where f denotes a face of the triangle; e denotes an edge of the triangle; $\vec{e}_{GT}$, $\vec{e}_r$,and $\vec{e}_f$ denote the edge vectors on f from M^i_{GT}, M^i_r and M^i_f, respectively. and n^f_{GT} denote the surface normal of f based on M^i_{GT}. The Laplace loss is introduced to maintain the smoothness of the local surface of the mesh:

$$L_{lap} = \sum_{i=1}^{N} \left\| \delta_i - \sum_{v_k \in N(v_i)} \delta_k / B_i \right\|_2^2, \tag{16}$$

where v_i and $\hat{v}_i$ denote the ground truth and estimated 2D/3D positions of the mesh vertices, $\delta_i = v_i^{3D} - \hat{v}_i^{3D}$ denotes the offset of the estimated value from the ground truth, $N(v_i)$ is the set of neighboring vertices of v_i, and B_i is the number of vertices in the set of $N(v_i)$. Our overall loss is $L = \lambda_m L_{mesh} + \lambda_j L_{joint} + \lambda_n L_{normal} + \lambda_e L_{edge} + \lambda_{lap} L_{lap}$. For the hyperparameters, we set λ_m=λ_j=λ_n=λ_e=1,λ_{lap}=50 in our implementation.

IV. EXPERIMENT

A. Datasets and evaluation metrics

InterHand2.6M dataset. The InterHand2.6M [3] dataset provides multiview RGB images with two-handed mesh and joint 3D annotations. We trained and evaluated LGNet in our experiments using a subset of 5 frames-per-second IH with H+M annotations.

HIC dataset. We further show the results on the HIC [27] dataset. Unlike the InterHand2.6M [3] dataset, which

has a uniform background and light sources, the HIC dataset contains more diverse backgrounds and natural light.The HIC dataset is used for evaluation only.

In-the-wild Datasets. We conducted qualitative experiments on the RGB2Hands dataset [20] and the EgoHands dataset [28]. These datasets contain complex interacting hand samples, diverse backgrounds, realistic lighting conditions, and varying image quality, which enable a comprehensive evaluation of the generalization ability of our method.

Evaluation metrics. We report the mean joint position error (MPJPE), mean vertex position error (MPVPE), and mean relative-root position error (MRRPE). All metrics are reported in millimeters. Additionally, we present the percentage of correct keypoints (PCK) curve and the area under the curve (AUC) across thresholds ranging from 0 to 50 millimeters to compare the reconstruction accuracy.

B. Implementation details

All implementations are done with PyTorch [29] using the Adam optimizer [30], with a batch size of 32 per GPU (trained with two RTX 3090 Ti GPUs). We trained the model for 30 epochs, with learning rate annealing at the 10th and 15th epochs from the initial learning rate of 1×10^{-4}. During training, we employ data augmentation techniques including scaling, rotation, random horizontal flipping, and color dithering.

C. Comparisons with state-of-the-art methods

TABLE I

QUANTITATIVE COMPARISON WITH STATE-OF-THE-ART METHODS ON THE INTERHAND2.6M [3] DATASET. † INDICATES SINGLE-HAND METHODS, ↓ INDICATES THE LOWER THE VALUE, THE BETTER.

Methods	MPJPE↓	MPVPE↓	MRRPE↓
†Zimmermann *et al*.[31]	36.36	-	-
†Zhou *et al*.[17]	23.48	23.89	-
†Boukhayma *et al*.[32]	16.93	17.98	-
†Spurr *et al*.[27]	15.40	-	-
InterNet[3]	16.01	-	32.60
DIGIT[1]	14.27	-	29.22
InterShape[4]	13.48	13.95	-
Keypoint Transformer[5]	12.78	-	29.63
IntagHand[6]	8.79	9.03	-
DIR[33]	7.51	7.72	28.98
ACR[34]	7.41	7.63	-
LGNet(Light.Ours)	**6.31**	**6.54**	32.18
LGNet(Ours)	**5.30**	**5.69**	**26.43**

Table I and Table II show that our LGNet achieves the highest performance on the InterHand2.6M [3] and HIC [27] datasets. In Table I, we compare the performance of our model with previous 3D interactive hand mesh reconstruction methods. Our LGNet shows significantly better performance. First, compared to the two-hand estimation method ACR[34], our method improves by 28.48% (5.30 mm vs. 7.41 mm), 25.43% (5.69 mm vs. 7.63 mm) on MPJPE and MPVPE. This indicates that our method demonstrates better pixel-aligned properties and better simulates the spatial relationship between the two hands. In Table II, our

TABLE II

QUANTITATIVE COMPARISON WITH STATE-OF-THE-ART METHODS ON THE HIC [27] DATASET.

Methods	MPVPE↓	MRRPE↓
Keypoint Transformer[5]	51.21	190.77
IntagHand[6]	45.74	-
InterShape[4]	42.08	-
LGNet(Light.Ours)	35.86	80.80
LGNet(Ours)	**31.79**	**78.47**

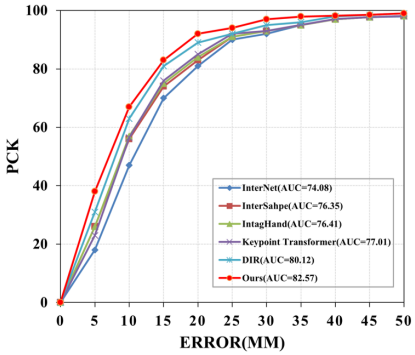

Fig. 6. Comparison with SOTA methods on InterHand2.6M.

LGNet achieves significantly better performance on the HIC dataset, indicating its strong generalization capability. The PCK curve in Figure 6 further demonstrates the superior performance of our method at all error threshold levels. Figure 7 demonstrates the robustness test of asymmetric hand poses [35]. Compare to the previous state-of-the-art method, LGNet shows significantly stronger robustness to asymmetric hand poses.

D. Qualitative Results

Our qualitative results on the InterHand2.6M [3] dataset are shown in Figure 8 and Figure 9. As shown in Figure 8, our method generates high-quality two-handed mesh reconstruction results in heavily occluded and complex interaction environments, and performs well even in a wide range of cases such as self-obscuration, close interaction, blurring, or occlusion. As shown in Figure 9, compared to the previous state-of-the-art method, our method produces more realistic finger interactions and fewer collisions between the two

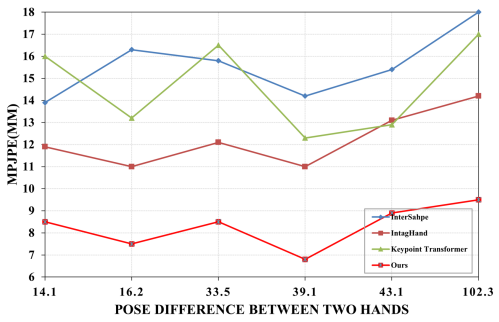

Fig. 7. Robustness to asymmetric poses of two hands on InterHand2.6M.

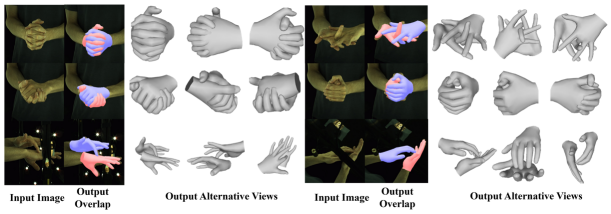

Fig. 8. Qualitative results of our method on the InterHand2.6M dataset [3].

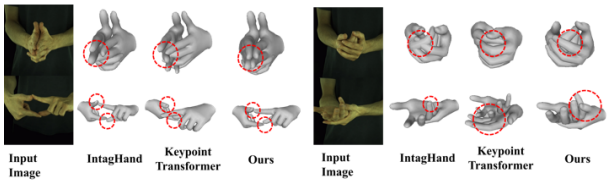

Fig. 9. Visual comparison with the state-of-the-art method [6], [5], [35] on the InterHand2.6M dataset [3].

hands. The results demonstrate the superior performance of our LGNet in accurately estimating hand poses and interactions. The results demonstrate the superior performance of our LGNet in accurate hand pose and interaction estimation. As shown in Figure 10, we demonstrate the generalization ability of our method on in-the-wild images. Due to its ability to utilize global information to enhance visual features, our method exhibits relative robustness to object perturbations and maintains hand structure effectively, showing significant advantages in preserving hand structure. Furthermore, our model achieves an inference speed of 30fps on a single NVIDIA RTX 3090 GPU, enabling potential real-time applications in the future.

E. Ablation study

TABLE III
COMPARING THE PERFORMANCE OF OUR FULL PIPELINE (BOTTOM-MOST) WITH VARIOUS ABLATION CASES.

Methods	MPJPE↓	MPVPE↓	MRRPE↓
w/o joint stage	10.04	9.36	30.37
w/o mesh stage	5.74	6.39	28.72
w/o refine stage	6.11	6.32	27.24
w/o L/R feature	7.09	6.21	27.46
w/o LGBlock	8.25	8.84	31.40
w/o Local unit	7.19	7.31	32.96
w/o Global unit	5.97	6.17	28.61
w/o L_{normal}	5.57	6.08	26.70
w/o L_{edge}	6.33	6.83	27.46
w/o L_{lap}	5.82	5.98	26.95
Ours(full)	**5.30**	**5.69**	**26.43**

The ablation results are reported in the Table III. We observed significant performance drops when removing any one component, indicating the effectiveness of each component and their contributions to the overall framework performance. Additionally, please refer to Figure 11 for visual comparison.

Fig. 10. Qualitative results on in-the-wild images.

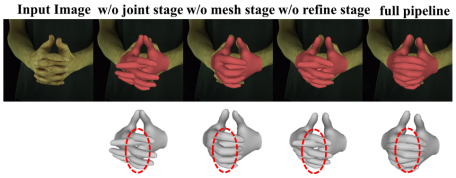

Fig. 11. Ablation study on our decoupled design.

V. CONCLUSION

We propose LGNet, a local and global feature adaptation network for recovering 3D interacting hand meshes from a single RGB image. Specifically, we decouple the 3D hand mesh reconstruction task into three stages: the joint stage predicts the 3D coordinates of hand joints; the mesh stage estimates a rough 3D hand mesh; and the refine stage helps to align the rough mesh with the hand image at the fingertip level. LGNet efficiently learns the interaction between two hands with the help of the main module LGBlock, which is mainly driven by our novel Transformer-based module LGFA, by extracting the interaction features from the features of the two hands using Local unit and Global unit, and then adapting the interaction features to each hand. Experimental results show that our method achieves state-of-the-art performance in recovering challenging 3D interactive hand meshes, which can serve as a baseline to inspire more research on arbitrary hand pose and shape reconstruction, and shows significant potential for applications in robotics.

REFERENCES

[1] Z. Fan, A. Spurr, M. Kocabas, S. Tang, M. J. Black, and O. Hilliges, "Learning to disambiguate strongly interacting hands via probabilistic per-pixel part segmentation," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 1–10.

[2] D. U. Kim, K. I. Kim, and S. Baek, "End-to-end detection and pose estimation of two interacting hands," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 189–11 198.

[3] G. Moon, S.-I. Yu, H. Wen, T. Shiratori, and K. M. Lee, "Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*. Springer, 2020, pp. 548–564.

[4] B. Zhang, Y. Wang, X. Deng, Y. Zhang, P. Tan, C. Ma, and H. Wang, "Interacting two-hand 3d pose and shape reconstruction from single color image," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 11 354–11 363.

[5] S. Hampali, S. D. Sarkar, M. Rad, and V. Lepetit, "Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 090–11 100.

[6] M. Li, L. An, H. Zhang, L. Wu, F. Chen, T. Yu, and Y. Liu, "Interacting attention graph for single image two-hand reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2761–2770.

[7] X. Di and P. Yu, "Lwa-hand: Lightweight attention hand for interacting hand reconstruction," in *European Conference on Computer Vision*. Springer, 2022, pp. 722–738.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[9] L. Ballan, A. Taneja, J. Gall, L. Van Gool, and M. Pollefeys, "Motion capture of hands in action using discriminative salient points," in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*. Springer, 2012, pp. 640–653.

[10] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Tracking the articulated motion of two strongly interacting hands," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 1862–1869.

[11] N. Kyriazis and A. Argyros, "Scalable 3d tracking of multiple interacting objects," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3430–3437.

[12] V. Choutas, G. Pavlakos, T. Bolkart, D. Tzionas, and M. J. Black, "Monocular expressive body regression through body-driven attention," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*. Springer, 2020, pp. 20–40.

[13] H. Joo, T. Simon, and Y. Sheikh, "Total capture: A 3d deformation model for tracking faces, hands, and bodies," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8320–8329.

[14] Y. Rong, T. Shiratori, and H. Joo, "Frankmocap: Fast monocular 3d hand and body motion capture by regression and integration," *arXiv preprint arXiv:2008.08324*, 2020.

[15] D. Xiang, H. Joo, and Y. Sheikh, "Monocular total capture: Posing face, body, and hands in the wild," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 10 965–10 974.

[16] Y. Zhang, Z. Li, L. An, M. Li, T. Yu, and Y. Liu, "Lightweight multi-person total motion capture using sparse multi-view cameras," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5560–5569.

[17] Y. Zhou, M. Habermann, I. Habibie, A. Tewari, C. Theobalt, and F. Xu, "Monocular real-time full body capture with inter-part correlations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4811–4822.

[18] J. Taylor, V. Tankovich, D. Tang, C. Keskin, D. Kim, P. Davidson, A. Kowdle, and S. Izadi, "Articulated distance fields for ultra-fast tracking of hands interacting," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, pp. 1–12, 2017.

[19] F. Mueller, M. Davis, F. Bernard, O. Sotnychenko, M. Verschoor, M. A. Otaduy, D. Casas, and C. Theobalt, "Real-time pose and shape reconstruction of two interacting hands with a single depth camera," *ACM Transactions on Graphics (ToG)*, vol. 38, no. 4, pp. 1–13, 2019.

[20] J. Wang, F. Mueller, F. Bernard, S. Sorli, O. Sotnychenko, N. Qian, M. A. Otaduy, D. Casas, and C. Theobalt, "Rgb2hands: real-time tracking of 3d hand interactions from monocular rgb video," *ACM Transactions on Graphics (ToG)*, vol. 39, no. 6, pp. 1–16, 2020.

[21] G. Moon, H. Choi, and K. M. Lee, "Neuralannot: Neural annotator for 3d human mesh training sets," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2299–2307.

[22] Y. Rong, J. Wang, Z. Liu, and C. C. Loy, "Monocular 3d reconstruction of interacting hands via collision-aware factorized refinements," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 432–441.

[23] H. Meng, S. Jin, W. Liu, C. Qian, M. Lin, W. Ouyang, and P. Luo, "3d interacting hand pose estimation by hand de-occlusion and removal," in *European Conference on Computer Vision*. Springer, 2022, pp. 380–397.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[26] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *arXiv preprint arXiv:2201.02610*, 2022.

[27] A. Spurr, J. Song, S. Park, and O. Hilliges, "Cross-modal deep variational hand pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 89–98.

[28] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1949–1957.

[29] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[31] C. Zimmermann and T. Brox, "Learning to estimate 3d hand pose from single rgb images," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4903–4911.

[32] A. Boukhayma, R. d. Bem, and P. H. Torr, "3d hand shape and pose from images in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 843–10 852.

[33] P. Ren, C. Wen, X. Zheng, Z. Xue, H. Sun, Q. Qi, J. Wang, and J. Liao, "Decoupled iterative refinement framework for interacting hands reconstruction from a single rgb image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8014–8025.

[34] Z. Yu, S. Huang, C. Fang, T. P. Breckon, and J. Wang, "Acr: Attention collaboration-based regressor for arbitrary two-hand reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 955–12 964.

[35] J. Park, D. S. Jung, G. Moon, and K. M. Lee, "Extract-and-adaptation network for 3d interacting hand mesh recovery," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4200–4209.